# Bash Shell Basics

**Bash stands for " Bourne again shell ". When you open a terminal window, it is bash that interprets the commands that you type into it and executes them for you.**

**Each instance of Bash(you can have multiple instances in multiple terminals) has a directory(or folder) referred to as the current directory. Think of it as your present location in the file system. Some of the Bash commands are listed below.**

**NOTE - These commands are some of the major commands that you need to memori But this doesn't form the exhaustive list of Bash Commands. There are some other commands like awk,sed,grep,etc. and other Networking Commands like telnet. Look them up on your own interest using the man pages (or) by Googling the commands Command to use the man pages are - "man command_name"**

**1) Commands to move around and explore the File System**

| COMMAND | DESCRIPTION |
|---|---|
| **pwd** | **Shows the current directory (present working directory)** |
| **cd** *dir_name* | **Change the current directory to** *dir_name* **(change directory)** |
| **cd ..** | **Change the current directory to the parent of the current directory** |
| **ls** | **List the files in the current directory along with subdirectories (excluding hidden files and hidden subdirectories)** |
| **ls -l** | **Longer form of ls which displays some details of each file in the present working directory** |
| **ls -a** | **List all files and subdirectories in the current working directory (including hidden files and hidden directories)** |
| **ls -la** | **Show all files including hidden files and hidden subdirectories in th longer form** |
| **cat** *file_name* | **Show the content of the file "***file_name***" on the screen** |
| **head** *file_name* | **Show the first 10 files of the file "***file_name"* **on the screen** |
| **tail** *file_name* | **Show the last 10 files of the file "***file_name"* **on the screen** |
| **wc** *file_name* | **Prints lines, words, and byte counts for** *file_name* |

**2) Commands to manipulate (create,delete,copy,move) files and directories**

| COMMAND | DESCRIPTION |
|---|---|
| **touch** *file1 file2 file3 . . . . .* | **Creates New Empty files** *file1,file2,file3 . . . .* **in the present directory (If files are already present, then their timestamps would be updated)** |
| **mkdir** *dir1 dir2 dir3 . . . .* | **Creates New Empty directories** *dir1,dir2,dir3 . . . .* **in the present directory** |
| **mkdir -p** *dir1/dir2/dir3/ . . . .* | Creates the directory *dir1* in the present directory. Creates *dir2* inside *dir1* ; *dir3* inside *dir2* and so on. |
| **cp** *file1 file2* | **Make a copy of file1 with the name** *file2* **in the same Directory** |
| **cp** *file1 dir1* | **Make a copy of** *file1* **inside the directory** *dir1* |
| **cp** *file1 file2 file3 . . . . . dir1* | **Make copies of** *file1,file2,file3,.....* **inside the directory** *dir1* **(Copy Multiple files into another Directory)** |
| **cp -r** *dir1 dir2* | **Copy the whole directory** *dir1* **into the directory** *dir2;* **Create** *dir2* **if it doesn't exist** |
| **mv** *file1 file2* | **Move** *file1* **to** *file2* **(Can be thought of as renaming)** |
| **mv** *file1 dir1* | **Move** *file1* **to the directory named** *dir1* |
| **mv** *dir1 dir2* | **Move** *dir1* **to the directory** *dir2* **(dir2 can be specified via Absolute Path a** |
| **rm** *file1 file2 file3 . . . . . .* | **Permanently Deletes the files** *file1,file2,file3 .....* |
| **rmdir** *dir1 dir2 dir3 . . . .* | **Permanently Deletes the Empty Directories** *dir1,dir2,dir3 . . . . .* **(The Directories** *dir1,dir2,dir3* **have to be empty, else this command won't work)** |
| **rm -rf** *file1/dir1* | **Permanently deletes the file** *file1* **(or) directory** *dir1* **(Deletes the directory** *dir1* **even if the directory isn't empty)** |

**3) Miscellaneous Commands**

| COMMAND | DESCRIPTION |
|---|---|
| **date** | **Prints the current time and date** |
| **whoami** | **Print the name of Current User who is logged in** |
| **who** | **List logged in users** |
| **w** | **Show what logged in users are doing** |
| **top** | **Show dynamic list of processes running. Type "q" to quit** |
| **vim** *file1* | **Edit** *file1* **using vim editor. Create** *file1* **if it does not exist** |

**4) Redirection and Piping**

| SYMBOL | DESCRIPTION |
|---|---|
| *Command* **>** *file1* | **Output of** *Command* **is placed into** *file1.* **( If** *file1* **doesn't exist, it is create If** *file1* **already exists then the previous content of** *file1* **is OVERWRITTE by the output of the** *Command* **)** |
| *Command* **<** *file1* | **Some commands need input from user (Like ./a.out) . By using the syml '<' the input is directly given by the content of the file, to the command Ex: ./a.out < testcase1 (testcase1 is a test case file)** |
| *Command* **>>** *file1* | '>>' redirects output to a file (just like <) appending the redirected output at the end, instead of overwriting the file *file1* , if it already exists |
| *Command1* **|** *Command2* | **'|' is the pipeline operator. The output of command1 is given as the inp of command2. This command can be extended to any number of pipeli like** *cmd1* | *cmd2* | *cmd3* | *cmd4* | . . . . . |

**5) Process Management**

| COMMAND | DESCRIPTION |
|---|---|
| **df** | **Show disk usage** |
| **du** | **Show directory space usage** |
| **ps** | **Display your currently active processes** |
| **top** | **Display all running processes** |
| **kill** *pid* | **Kill the process with process id** *pid* |
| **Kill -9** *pid* | **Force Kill the process with process id** *pid* **instantly** |
| **killall** *proc* | **Kill all processes named proc *** |
| **bg** | **Lists stopped or background jobs; resume a stopped job in the background** |
| **fg** | **Brings the most recent job to foreground** |
| **fg** *proc* | **Brings job** *proc* **to the foreground** |

### 6) Basic Server Commands

| COMMAND | DESCRIPTION |
|---|---|
| **ssh** *user@host* | **connect to** *host* **as** *user* |
| **ssh -p** *port user@host* | **connect to** *host* **on port** *port* **as** *user* |
| **ssh-copy-id** *user@host* | **Add your key to host for user to enable a keyed or passwordless lo** |
| **scp** *user@host:file1 dirpath* | **Copies** *file1* **from remote machine to the directory specified(***dirpat* **into local system** |
| **scp** *file1 user@host:dirpath* | **Copies** *file1* **from local system to the directory ,***dirpath,* **present in t remote machine** |
| **scp -r** *dir1 user@host:dir2* | **Copies the whole directory contents ,***dir1***, present in the local syste to the directory** *dir2* **of remote machine ; Similarly this can be appli to transfer directory contents from remote machine to local syste** |
| **wget** *url* | **Downloads the web content specified by the url, into the present working directory** |

### 7) Searching Commands

| COMMAND | DESCRIPTION |
|---|---|
| **grep** *pattern file* | **search for** *pattern* **in** *file* |
| **grep -r** *pattern dir* | **search recursively for** *pattern* **in** *dir* |
| *command* **\| grep** *pattern* | **search for** *pattern* **in the output of** *command* |
| **locate** *file1* | **find all instances of** *file1* |

### 8) Changing File Permissions

**Please refer the below link for the complete understanding of what file permissions are and how to mod them.**

**https://www.tutorialspoint.com/unix/unix-file-permission.htm**